


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

dynamic multithreaded processor

SEARCH


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **dynamic multithreaded processor**

Found 15,944 of 131,734

Sort results by

relevance

Display results

expanded form

[Save results to a Binder](#)[Search Tips](#)☐ Open results in a new windowTry an [Advanced Search](#)Try this search in [The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1 [A dynamic multithreading processor](#)**

Haitham Akkary, Michael A. Driscoll

November 1998 **Proceedings of the 31st annual ACM/IEEE international symposium on Microarchitecture**Full text available: [pdf\(2.67 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**2 [A survey of processors with explicit multithreading](#)**

Theo Ungerer, Borut Robič, Jurij Šilc

March 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 1Full text available: [pdf\(920.16 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Hardware multithreading is becoming a generally applied technique in the next generation of microprocessors. Several multithreaded processors are announced by industry or already into production in the areas of high-performance microprocessors, media, and network processors. A multithreaded processor is able to pursue two or more threads of control in parallel within the processor pipeline. The contexts of two or more threads of control are often stored in separate on-chip register sets. Unused i ...

Keywords: Blocked multithreading, interleaved multithreading, simultaneous multithreading

3 [Improving server software support for simultaneous multithreaded processors](#)

Luke K. McDowell, Susan J. Eggers, Steven D. Gribble

June 2003 **ACM SIGPLAN Notices , Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 38 Issue 10Full text available: [pdf\(218.63 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Simultaneous multithreading (SMT) represents a fundamental shift in processor capability. SMT's ability to execute multiple threads simultaneously within a single CPU offers tremendous potential performance benefits. However, the structure and behavior of software affects the extent to which this potential can be achieved. Consequently, just like the earlier arrival of multiprocessors, the advent of SMT processors prompts a needed re-evaluation of software that will run on them. This evaluation ...

Keywords: runtime support, servers, simultaneous multithreading

4 Clustered speculative multithreaded processors

Pedro Marcuello, Antonio González

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available:  pdf(1.17 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: clustered processors, control-flow speculation, data dependence speculation, data value speculation, dynamically scheduled processors, simultaneous multithreaded processors

5 Speculative multithreaded processors

Pedro Marcuello, Antonio González, Jordi Tubella

July 1998 **Proceedings of the 12th international conference on Supercomputing**

Full text available:  pdf(1.24 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: control speculation, data dependence speculation, data speculation, dynamically scheduled processors, multithreaded processors

6 Implicitly-multithreaded processors

Il Park, Babak Falsafi, T. N. Vijaykumar

May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture**, Volume 31 Issue 2

Full text available:  pdf(376.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

This paper proposes the Implicitly-MultiThreaded (IMT) architecture to execute compiler-specified speculative threads on to a modified Simultaneous Multithreading pipeline. IMT reduces hardware complexity by relying on the compiler to select suitable thread spawning points and orchestrate inter-thread register communication. To enhance IMT's effectiveness, this paper proposes three novel microarchitectural mechanisms: (1) resource- and dependence-based fetch policy to fetch and execute suitable ...

7 Slipstream processors: improving both performance and fault tolerance

Karthik Sundaramoorthy, Zach Purser, Eric Rotenberg

November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 28 , 34 Issue 5 , 5

Full text available:  pdf(111.54 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Processors execute the full dynamic instruction stream to arrive at the final output of a program, yet there exist shorter instruction streams that produce the same overall effect. We propose creating a shorter but otherwise equivalent version of the original program by removing ineffectual computation and computation related to highly-predictable control flow. The shortened program is run concurrently with the full program on a chip multiprocessor simultaneous multithreaded processor, with two ...

8 Slipstream processors: improving both performance and fault tolerance

Karthik Sundaramoorthy, Zach Purser, Eric Rotenberg

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11


Full text available:  [pdf\(1.51 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Processors execute the full dynamic instruction stream to arrive at the final output of a program, yet there exist shorter instruction streams that produce the same overall effect. We propose creating a shorter but otherwise equivalent version of the original program by removing ineffectual computation and computation related to highly-predictable control flow. The shortened program is run concurrently with the full program on a chip multiprocessor or simultaneous multithreaded processor, with t ...

9 Tolerating memory latency through software-controlled pre-execution in simultaneous multithreading processors

Chi-Keung Luk

May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture**, Volume 29 Issue 2

Full text available:  [pdf\(1.11 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

Hardly predictable data addresses in many irregular applications have rendered prefetching ineffective. In many cases, the only accurate way to predict these addresses is to directly execute the code that generates them. As multithreaded architectures become increasingly popular, one attractive approach is to use idle threads on these machines to perform pre-execution—essentially a combined act of speculative address generation and prefetching—to accelerate the main thread ...

10 A study of slipstream processors

Zach Purser, Karthik Sundaramoorthy, Eric Rotenberg



December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  [pdf\(130.26 KB\)](#)
 [ps\(398.01 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)
 [Publisher Site](#)

11 Dynamically allocating processor resources between nearby and distant ILP

Rajeev Balasubramanian, Sandhya Dwarkadas, David H. Albonesi

May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture**, Volume 29 Issue 2

Full text available:  [pdf\(998.02 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
 [Publisher Site](#)



Modern superscalar processors use wide instruction issue widths and out-of-order execution in order to increase instruction-level parallelism (ILP). Because instructions must be committed in order so as to guarantee precise exceptions, increasing ILP implies increasing the sizes of structures such as the register file, issue queue, and reorder buffer. Simultaneously, cycle time constraints limit the sizes of these structures, resulting in conflicting design requirements.

In ...

12 Multithreading and value prediction: Dynamic speculative precomputation

Jamison D. Collins, Dean M. Tullsen, Hong Wang, John P. Shen

December 2001 **Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  [pdf\(1.37 MB\)](#)  Additional Information:

[Publisher Site](#)[full citation](#), [abstract](#), [references](#), [citations](#)

A large number of memory accesses in memory-bound applications are irregular, such as pointer dereferences, and can be effectively targeted by thread-based prefetching techniques like Speculative Precomputation. These techniques execute instructions, for example on an available SMT thread context, that have been extracted directly from the program they are trying to accelerate. Proposed techniques typically require manual user intervention to extract and optimize instruction sequences. This paper ...

13 [Control independence in trace processors](#)

Eric Rotenberg, Jim Smith

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**

Full text available: [pdf\(1.40 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

Branch mispredictions are a major obstacle to exploiting instruction-level parallelism, at least in part because all instructions after a mispredicted branch are squashed. However, instructions that are control independent of the branch must be fetched regardless of the branch outcome, and do not necessarily have to be squashed and re-executed. Control independence exists when the two paths following a branch re-converge. A trace processor ...

14 [An analysis of operating system behavior on a simultaneous multithreaded architecture](#)

Joshua A. Redstone, Susan J. Eggers, Henry M. Levy

November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 28, 34 Issue 5, 5

Full text available: [pdf\(227.80 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents the first analysis of operating system execution on a simultaneous multithreaded (SMT) processor. While SMT has been studied extensively over the past 6 years, previous research has focused entirely on user-mode execution. However, many of the applications most amenable to multithreading technologies spend a significant fraction of their time in kernel code. A full understanding of the behavior of such workloads therefore requires execution and measurement of the operating system ...

15 [An analysis of operating system behavior on a simultaneous multithreaded architecture](#)

Joshua A. Redstone, Susan J. Eggers, Henry M. Levy

November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11

Full text available: [pdf\(1.56 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper presents the first analysis of operating system execution on a simultaneous multithreaded (SMT) processor. While SMT has been studied extensively over the past 6 years, previous research has focused entirely on user-mode execution. However, many of the applications most amenable to multithreading technologies spend a significant fraction of their time in kernel code. A full understanding of the behavior of such workloads therefore requires execution and measurement of the operating system ...

16 [Symbiotic jobscheduling for a simultaneous multithreaded processor](#)

Allan Snaveley, Dean M. Tullsen

November 2000 **Proceedings of the ninth international conference on Architectural support for programming languages and operating systems**, Volume 34, 28 Issue 5, 5

Full text available: [pdf\(247.10 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Simultaneous Multithreading machines fetch and execute instructions from multiple instruction streams to increase system utilization and speedup the execution of jobs. When there are more jobs in the system than there is hardware to support simultaneous execution, the operating system scheduler must choose the set of jobs to coschedule. This paper demonstrates that performance on a hardware multithreaded processor is sensitive to the set of jobs that are coscheduled by the operating system scheduler ...

17 Value prediction for speculative multithreaded architectures

Pedro Marcuello, Jordi Tubella, Antonio González

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  pdf(882.88 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

 [Publisher Site](#)

The speculative multithreading paradigm (speculative thread-level parallelism) is based on the concurrent execution of control-speculative threads. The efficiency of microarchitectures that adopt this paradigm strongly depends on the performance of the control and data speculation techniques. While control speculation is used to predict the most effective points where a thread can be spawned, data speculation is required to eliminate the serialization imposed by inter-thread dependence ...


18 Multithreading I: Master/slave speculative parallelization

Craig Zilles, Gurindar Sohi

November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

Full text available:  pdf(1.31 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

 [Publisher Site](#)

Master/Slave Speculative Parallelization (MSSP) is an execution paradigm for improving the execution rate of sequential programs by parallelizing them speculatively for execution on a multiprocessor. In MSSP, one processor---the master---executes an approximate version of the program to compute selected values that the full program's execution is expected to compute. The master's results are checked by slave processors that execute the original program. This validation is parallelized by cutting ...

19 Embedded systems: applications, solutions and techniques (EMBS): Fine-grained power management for multithreaded processor cores

Sascha Uhrig, Theo Ungerer

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available:  pdf(225.41 KB)

Additional Information: [full citation](#), [abstract](#), [references](#)

We propose a new hardware-based power management technique that is made possible by a multithreaded processor core. A processor-internal scheduler manages frequency and voltage scaling based on the current processor utilization given in percentage of the total performance.

Keywords: multithreading, performance adaptation, power-aware program execution, power-management

20 Exploiting choice: instruction fetch and issue on an implementable simultaneous multithreading processor

Dean M. Tullsen, Susan J. Eggers, Joel S. Emer, Henry M. Levy, Jack L. Lo, Rebecca L. Stamm

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available:  pdf(1.48 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Simultaneous multithreading is a technique that permits multiple independent threads to issue multiple instructions each cycle. In previous work we demonstrated the performance potential of simultaneous multithreading, based on a somewhat idealized model. In this paper we show that the throughput gains from simultaneous multithreading can be achieved *without* extensive changes to a conventional wide-issue superscalar, either in hardware structures or sizes. We present an architecture for s ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)